Lecture Notes in Management Science (2016) Vol. **8**, 47–53 ISSN 2008-0050 (Print), ISSN 1927-0097 (Online)

A branch-and-price algorithm for dynamic sector configuration

Tambet Treimuth [1], Daniel Delahaye [1], Sandra Ulrich Ngueveu [2,3]
[1] French Civil Aviation University (ENAC) F-31400 Toulouse, France
{treimuth, delahaye}@recherche.enac.fr
[2] University of Toulouse, INP, LAAS, F-31400 Toulouse, France
[3] CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France
ngueveu@laas.fr

Proc. ICAOR 2016
Rotterdam, The Netherlands

Keywords:
Air traffic management
Airspace sectorization
Branch and price
Column generation

Abstract

Air traffic generates workload for the air traffic controllers in charge of the airspace. For a large airspace, a single air traffic controller is not able to manage all this workload and the airspace is divided into sectors, each of them being assigned to a controller. When the traffic demand is decreasing during the night, the sectors are gathered together into groups to reduce the number of controllers in operation. Nowadays, this regrouping is performed empirically by airspace experts. In this paper, we show how the branch-and-price method can be used to compute a balanced grouping of air traffic control sectors to optimally reduce the number of controller teams during daily low flow periods.

Introduction

Airspace is divided into sectors each of them being assigned to a controller team. For instance, the French airspace is divided into 89 sectors with different altitude extensions. Over the course of a day of normal traffic, the control workload fluctuates based on traffic demands between various origin-destination pairings. Traffic demand is reduced at night inducing less workload in the airspace and therefore sectors are gathered together into groups, each group constituting a larger sector assigned to a controller. The objective of the algorithm developed is to adapt the airspace according to the dynamic of traffic. In order to reach this objective, we consider that airspace is divided into elementary sectors which have to be merged or split with time. Each sector having a workload, we are searching for groupings with balanced workloads between groups of sectors and this, for each time period. When an aircraft is transferred from one sector to another, there is some information exchanges between the controllers concerned and the pilots in order to ensure a safe transfer of the aircraft between sectors. This information exchange between controllers and pilots induce an additional coordination workload for controller which is linked to the flow cut by the sector borders, related to the number of flights that cross sector borders. In order to reduce the transfer workload, the algorithm will have to minimize the flow cuts as well. Sector design requires also that sectors have to be connected when belonging to the same group. In the following of this paper, the problem of finding dynamic grouping of airspace sectors will be called Dynamic Sector Configuration (DAC) and a sector configuration will be referred to as *configuration*.

Previous work

DAC has a number of conflicting objectives, related to either the static (coordination, sector control and transition workloads) or the dynamic aspects of the problem. Therefore, multi-objective approaches have been suggested as a method for the problem (Brinton *et al*., 2008). Some 2D DAC optimization approaches have been developed, such as constraint programming (Trandac *et al*, 2003) and geometric algorithms (Basu *et al*., 2008), but these did not consider the multi-objective aspect of the problem. There exists also genetic programming based methods (Delahaye *et al*, 2008; Chen *et al*, 2013) (Xue *et al*, 2008) improved the Genetic Algorithm efficiency with an iterative algorithm. Initial works were limited to 2D design but recent works include the third dimension (Sergeeva *et al*, 2015). (Tang *et al*, 2012). proposed an improved agent-based model (iABM) combined with GA to achieve the optimized sectorization. Authors of (Li *et al*, 2009) use a spectral clustering based algorithm and (Kulkarni *et al*, 2011) has proposed an algorithm based on Dynamic Programming. Those previous work address partially the full problem and do not fully take into account the stability of the design between transitions. Indeed, air traffic controllers have to adapt to the changes which translate into extra workload. Therefore, transitions between configurations have to maintain some level of stability with time.

The remainder of the paper is organised as follows. The next section presents the DAC problem statement. Then a set partitioning-based mathematical formulation is introduced. The Branch and Price approach is described in the following section. Finally, computational results are presented before the Conclusion.

Problem statement

Let $T=\{1,...,|T|\}$ represent the discretized time horizon. The airspace is modelled with a time-dependent weighted graph $G=(V,E,W^t,t\in T)$ where V is the set of vertices, E is the set of edges and W^t is the set of edge weights and vertices weights at time period $t\in T$. Each vertex $i\in V$ represents a (elementary) sector i of the airspace and its weight corresponds to the sector workload. Each edge ($e=(i,j)$) represents the border between two neighbour sectors (i and j). Its weight corresponds to the coordination workload for the planes crossing the border if the two sectors belong to two different controllers. Our goal is to define a valid airspace configuration for each time period $t\in T$. A solution of DAC is therefore a graph partitioning of G for each time period t, where nodes belonging to the same partition at time period t correspond to sectors grouped together and assigned to the same controller at that time period.

We consider three main objectives to minimize: (a) the air traffic controllers' workload, (b) the workload difference between controllers, and (c) the changes between consecutive configurations. Therefore, the *static* cost of a solution is covered by the first two objectives and measures the quality of the sector grouping of each time period. The *dynamic* cost of a solution is covered by the third objective and measures the change rate between consecutive configurations. The user can therefore define his unique objective-function as a weighted sum of the three objectives $(\alpha a+\beta b+\gamma c)$, using γ to increase or decrease the relative importance of stability (the higher γ, the less changes will be favoured among consecutive sectorizations).

Set-partitioning-based mathematical formulation

The mathematical model proposed is based on the definition of the set of feasible configurations C^t for each time period t. These sets are of course exponential in size, hence the Dantzig-Wolfe decomposition and column-generation-based solution method. Each configuration i is assigned a cost denoted c_i^t which reflects its static cost if it is applied at time period t. An edge $e\in E$ is said to be frontier for configuration i if and only if its extremities belong to different controllers in the configuration i. An edge is said to be frontier at time period t if it is frontier in the configuration applied at time period t.

We introduce three sets of binary decision variables to model the DAC problem using an extended formulation: X_i^t, Y_e^t and Z_e^t. Variable X_i^t is equal to 1 if configuration $i\in C^t$ is applied at time period t and 0 otherwise. Variable Y_e^t is equal to 1 if edge $e\in E$ is a frontier at time t and 0 otherwise. Variable Z_e^t is equal to 1 if edge $e\in E$ was not frontier at time t-1 but became frontier at time t, or if edge e was frontier at time t-1 and is no longer frontier at time t. This variable is equal to 0 if the edge status (frontier or not) has not changed from t-1 to t. The resectorization problem can be formulated as follows:

$$(P_{CFB})\min f=\sum_{t=1}^{T}\sum_{i\in C^t}c_i^t X_i^t+\gamma\sum_{t=1}^{T}\sum_{e\in E}Z_e^t \tag{1}$$

s.t.

$$\sum_{i\in C^t}X_i^t=1, \qquad \forall t\in\{1..T\} \tag{2}$$

$$-Y_e^t+\sum_{i\in C^t(e)}X_i^t=0, \qquad \forall t\in\{1..T\},\forall e\in E \tag{3}$$

$$Z_e^t-Y_e^t+Y_e^{t-1}\geq 0, \qquad \forall t\in\{2..T\} \tag{4}$$

$$Z_e^t-Y_e^{t-1}+Y_e^t\geq 0, \qquad \forall t\in\{2..T\} \tag{5}$$

$$Y_e^t\in\{0,1\}, \qquad \forall t\in\{1..T\},\forall e\in E \tag{6}$$

$$X_i^t \in \{0,1\}, \qquad \forall t \in \{1..T\}, \forall i \in C^t \qquad (7)$$

$$Z_e^t \in \{0,1\}, \qquad \forall t \in \{2..T\}, \forall e \in E \qquad (8)$$

where γ is a predefined dynamic/static weighting factor and $C^t(e)$ is the subset of configurations valid for time t that use edge e as a frontier.

The objective-function (1) minimizes the sum of the static and dynamic costs. The static cost measures the quality of the configuration with regards to the traffic at the time it is applied, whereas the dynamic cost measures the stability of configurations over time. The static cost is computed as a weighted sum of the workload difference among controllers and the total amount of coordination workload. Indeed some configurations might be balanced but generate a higher overall workload for coordination. The dynamic cost is proportional to the number of edges that change status between two consecutive time periods. Constraints (2) ensure that one feasible configuration is applied per time period. Constraints (3) link the status of each edge (frontier or not) with the set of configurations ($C^t(e)$) in which the edge is a frontier. Constraints (4) and (5) ensure that the changes of status between consecutive instants are correctly computed for each edge.

This model has a polynomial number of binary variables Y_e^t and Z_e^t, but an exponential number of binary variables X_i^t. Let (P') be the problem obtained by replacing constraints (7) and (8) with (9) and (10) has the same optimal solution as (P). Its validity is stated by Theorem 1.

$$0 \le X_i^t, \qquad \forall t \in \{1..T\}, \forall i \in C^t \qquad (9)$$

$$0 \le Z_e^t \le 1, \qquad \forall t \in \{2..T\}, \forall e \in E \qquad (10)$$

Theorem 1. *An optimal solution of MILP (P') provides an optimal solution for problem (P).*

Proof outline: Although variables X_i^t are continuous, the model ensures that their values are always binary in feasible solutions, equal to 1 if configuration i is being applied at time t, and 0 otherwise. The same needs to be proven for variables Z_e^t.

Proving that equations (4), (5), (6), (10) and objective-function (1) result into $Z_e^t=0$ *if* $Y_e^{t-1}=Y_e^t$ and $Z_e^t=1$ otherwise, is straightforward. Regarding X_i^t, let $\underline{S}$ be a feasible solution of (P'): $\underline{Y} \in [0,1]^{|C||T|}$, $\underline{Z} \in [0,1]^{|E||T|}$ and $\underline{X} \in IR^{|L||T|}$. Given an instant $t^* \in T$, let us denote:

- $C^{>0}$ the subset of configurations used at instant t^*. In other words $C^{>0}=\{i : \underline{X}_i^{t*} > 0, \ \forall i \in C^{t*}\}$.
- $E^{>0}$ the subset of edges that are frontier in at least one configuration i of $C^{>0}$.
- $C^{>0}(e)$ the subset of configurations from $C^{>0}$ that use edge e as a frontier.

With a reasoning similar to the one applied for the extended formulation of (Ngueveu *et al*, 2016), it can be proven that $C^{>0}(e) = C^{>0}, \ \forall e \in E^{>0}$, from which it can be deduced that:

- either $|C^{>0}|=1$ and therefore all X_i^{t*} are binary
- or $|C^{>0}|>1$ but all configurations from $C^{>0}$ are identical.

There are no identical configurations in C^t of (P') therefore X_i^t are binary and (P') is equivalent to (P).

Branch-and-Price for DAC

Problem (P') has a polynomial number of constraints but an exponential number of continuous variables and some binary variables, therefore column generation can be applied to solve its linear relaxation and a branch-and-price solution method can be applied to solve (P'). Note that (P) could also be solved with a branch-and-price. However, allowing variables X_i^t to be continuous as it is done in (P') does ensure that the optimal solution can be obtained without needing to branch on the variables that are being generated by the pricing. As a consequence, the pricing procedure proposed remains valid at any node of the exploration tree. This means that the branching procedure can be left up to the mixed integer linear solver. Only the pricing procedure needs to be specifically implemented.

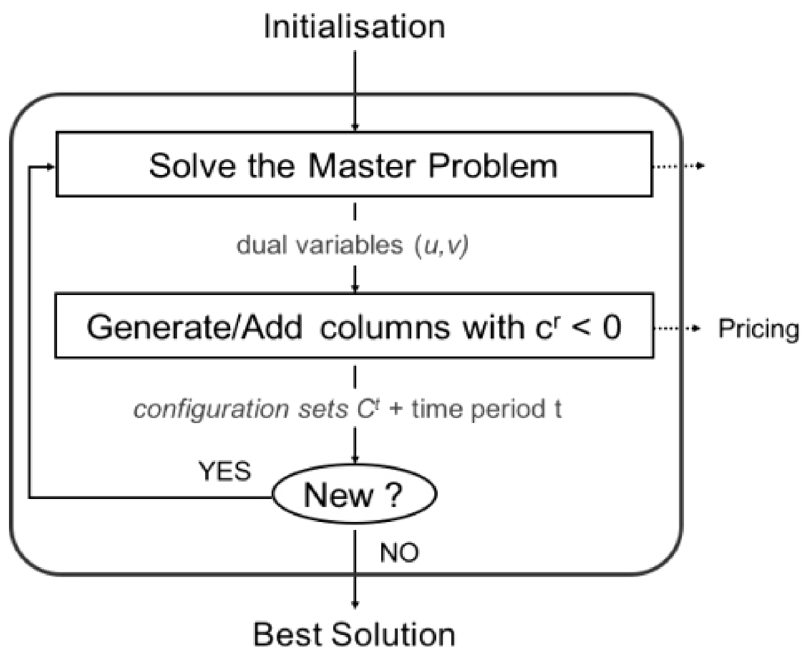

Figure 1. Column generation scheme

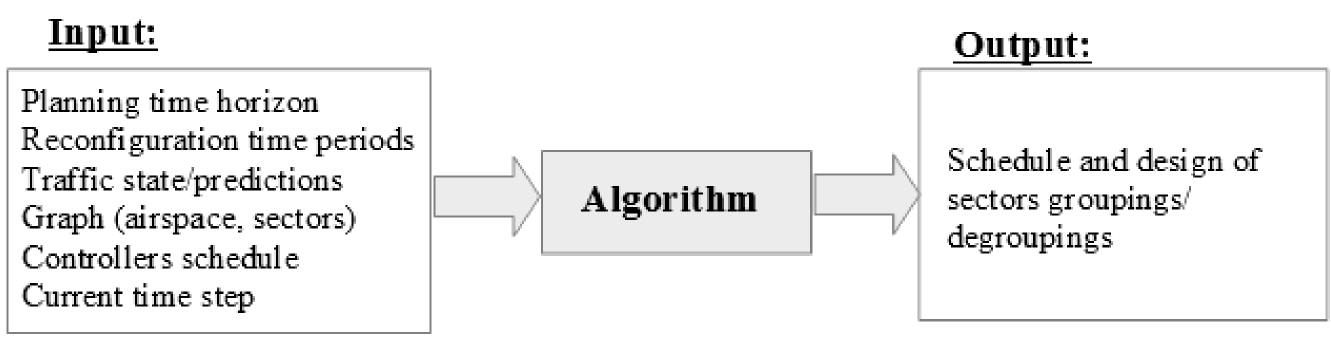

Figure 2. Dynamic sector configuration algorithm

Column generation for DAC

A master problem (P^r) for DAC is a linear relaxation of (P'), where the set of valid configurations C^t is replaced with a subset $C^{tr} \subset C^t$. Each master problem can therefore be solved with any linear programming solver. Let us consider the dual of (P') (resp. (P^r)) denoted (D')(resp. (D^r)). Let u^t, $\forall t \in \{1..T\}$ be the dual variables associated with constraints (12) and let v_e^t, $\forall t \in \{1..T\}$, $\forall e \in E$ be the dual variables associated with constraints (13). Any variable X_i^t in (P') corresponds to a dual constraint in (D') expressed with equation (14).

$$u_t + \sum_{e \in F(i)} v_e^t \le c_i^t \tag{14}$$

where *F(i)* is the set of edges which are frontier in configuration *i*.

Let $j \in C^t$ be a feasible configuration that does not belong to the restricted subset C^{tr}, then the corresponding variable X_j^t does not exist in (P^r) and the corresponding dual constraint (14) *j* does not exist in (D^r). Let S_{Dr} be the optimal solutions of (D^r). Only one of two different situations can happen: either constraint (14) *j* is violated by S_{Dr}, or it is not. If the constraint is not violated, then adding it would not change the optimum dual solution, which means that adding primal variable X_i^t would not change the optimal solution of (P_r). There is therefore no need to add configuration *j* in C^{tr}. If the constraint is violated, then adding it would change the optimal dual solution, meaning that the corresponding variable X_i^t should be added to the master problem and the configuration *j* should be added to C^{tr}. Consequently, our pricing procedure searches for missing and violated constraints (14), which is equivalent to searching for configurations of negative reduced cost where the reduced cost of a configuration *i* is computed with expression $c_i^t - u^t - \sum_{e \in E(i)} v_e^t$. The major advantage of using an extended formulation for modelling and solving DAC is to let the model decide optimally what is the best selection and sequence of configurations, while focussing on generating for each time period the configurations of negative reduced cost. Note that this reduced cost takes into account both the static and the dynamic impact of the addition of the configuration at time period *t*, by including the values of c_i^t and v_e^t in its computations.

Pricing procedure for DAC: how to find configurations of negative reduced cost

Given a weighted graph H representing the airspace at a specific time period, there exists different graph partitioning algorithms that can be used to find the best configuration for that specific time period. Our pricing method modifies the graph weights before using the partitioning algorithm to ensure that the configurations generated are the ones with the best reduced costs. Recall that the reduced cost of a configuration is given by expression: $c_i^t - u^t - \sum_{e \in E(i)} v_e^t$.

For a given time period *t,* u^t is constant and has the same value for all configurations, therefore finding the configuration that has the best reduced cost at time *t* is equivalent to finding the configuration *i* with the best $c_i^t - \sum_{e \in F(i)} v_e^t$ value, and then substracting u^t. Let α and β be the weights assigned to the controllers' workload difference (δ_i^t) and the edge costs in the computation of the static cost of a configuration, i.e.

$$c_i^t = \alpha \Delta \delta_i^t + \beta \sum_{e \in F(i)} w_e^t$$

then,

$$c_i^t - \sum_{e \in F(i)} v_e^t = \alpha \Delta \delta_i^t + \beta \sum_{e \in F(i)} w_e^t - \sum_{e \in F(i)} v_e^t = \alpha \Delta \delta_i^t + \beta \sum_{e \in F(i)} (w_e^t - \frac{v_e^t}{\beta})$$

As a consequence, finding the configuration that has the best reduced cost on the graph G with edge costs w_e is equivalent to finding the configuration that has the minimum total cost on graph H with modified edge costs $w_e^{tr} = w_e^t - v_e^t/\beta$ and then subtracting u^t from that total cost. Therefore, it suffices to apply classic graph partitioning on the graph with modified edge costs w_e^{tr} and then subtracting u^t from the total cost (on the modified graph) of the configuration obtained. If a negative value is obtained then a configuration of negative reduced cost has been found and should be added to the restricted set; otherwise, the current solution is optimal and the column generation stops.

Any airspace sectorization algorithm based on graph partitioning is applicable on the modified graph in the pricing procedure. This is another major advantage of the method proposed, because any additional restriction or regulation could be simply integrated in this pricing. Note that if the pricing procedure is heuristic, the resulting branch-and-price code is also heuristic, yet valid bounds and feasible solutions of good quality can be obtained. If the pricing procedure is an exact method, then the optimality of the final solution is guaranteed for the DAC.

Computational results

The proposed algorithm was tested on instances from the Reims ATCC. The real radar data was used for those computations. In Reims, the sectorization can be modified every 15 minutes and the planning has to be done for the next 2 hours, therefore $T = 8$ time periods. This computation was done at 6 different times: 8h00, 8h15, …9h15, resulting into 6 time steps. For the subsequent computational evaluation, we used a Multi-Level heuristic (Bichot *et al*, 2011) in the pricing procedure, which is fast yet efficient. The solutions computed were compared with the configurations used in the real situation in terms of (a) load balance between controllers and (b) smoothness (stability) of the sectorizations. The results showed that the proposed algorithm produced solutions with a better workload balance [Figure 3]. It means that controllers have more similar workload. The solutions were also better from a smoothness viewpoint. This means that consecutive sectorizations were more similar with proposed algorithm than the sectorizations that were used in practice [Figure 4].

The initial master problem was created with |T|=8 initial columns (configurations), one per time period, generated by applying the pricing procedure on the original weighted graph for each time period with all dual variables values set to zero. Then the branch-and-price procedure was launched and new columns (configurations) were generated until no configuration of negative reduced cost could be found. The algorithm was run at each of the 6 times steps. [Table 1] shows the computing times with an implementation in C++ and with the framework SCIP 3.1.0 on an Intel Core i5-4210U CPU and 6 GB of RAM. Each master problem was solved with IBM CPLEX 12.6 whereas each pricing was done with the Multi-Level heuristic.

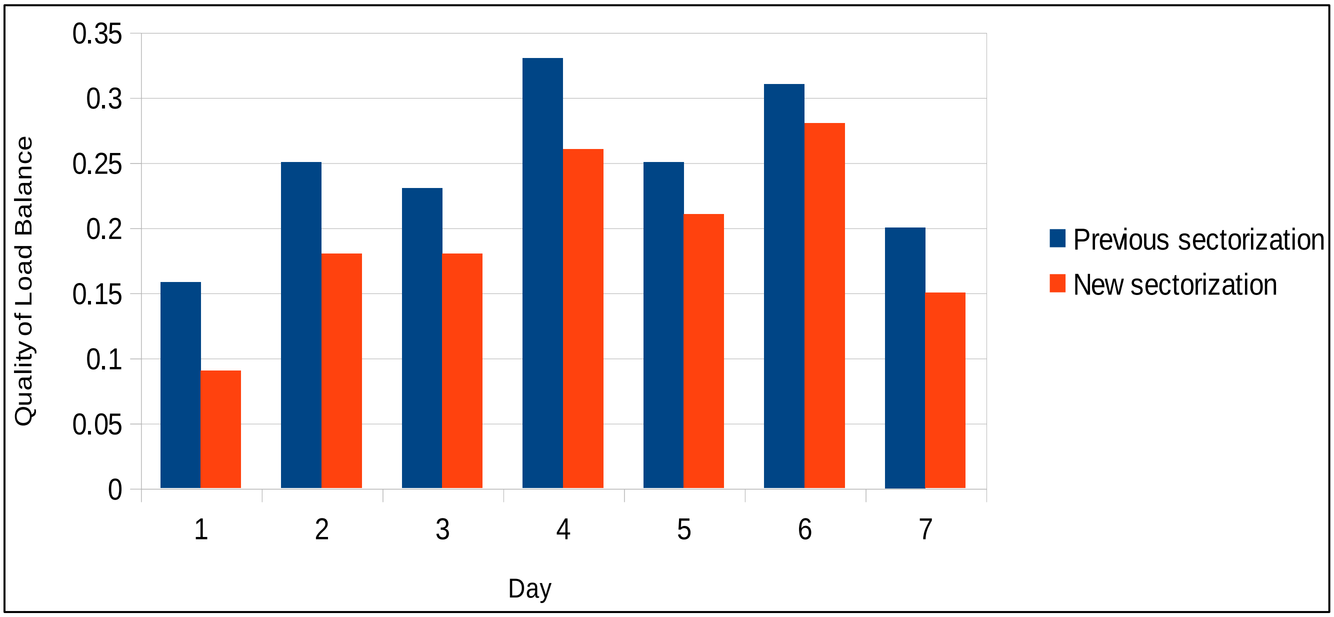

Figure 3. Quality of Load Balance

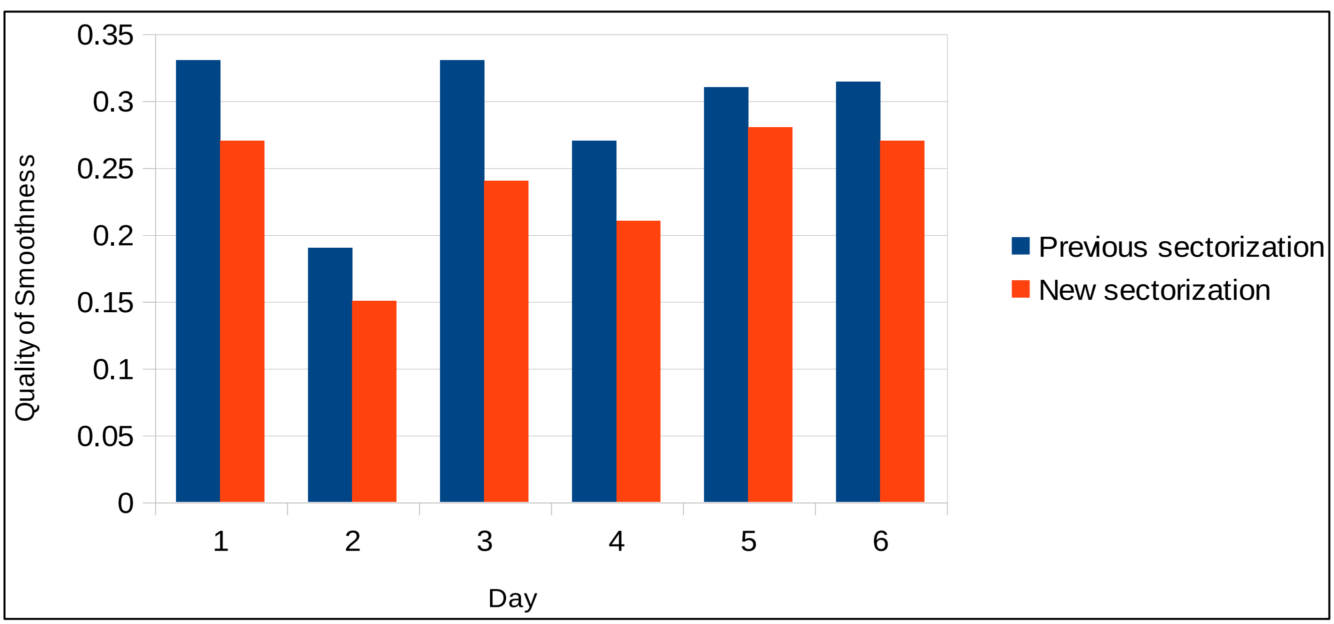

Figure 4. Quality of Smoothness

Table 1. Performance of the algorithm (SCIP, C++) at the different time

Time-steps	*1 (8h00)*	*2 (8h15)*	*3 (8h30)*	*4 (8h45)*	*5 (9h00)*	*6 (9h15)*
# configurations generated	55	48	40	35	72	45
Computing time	175 s	155 s	130 s	119 s	201 s	145 s

Conclusion

This paper presented a branch-and-price-based method for solving an airspace dynamic sector configuration problem. The method proposed is generic, adaptive and was fast on real world instances. It allows the user to define his unique objective function by choosing the relative weight of the static versus the dynamic aspects of the problem: controller workload balance versus stability of the sectorization. It can therefore be used for an automatic planning of sectorization, to decide when to reconfigure the airspace and how often. Future work includes the automation of the parameter setting procedure that fine-tunes the weighting coefficients in function of the user expressed preferences.

References

Basu, A., Mitchell, J. S., & Sabhnani, G. (2008). Geometric Algorithms for Optimal Airspace Design and Air Traffic Controller Workload Balancing. 2008 Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments (ALENEX), 75-89.

Bichot,C.,& Siarry, P.(2011). Graph partitioning. London:ISTE.

Brinton, C. R., & Pledgie, S. (2008). Airspace partitioning using flight clustering and computational geometry. IEEE/AIAA 27th Digital Avionics Systems Conference.

Chen, Y., Bi, H., Zhang, D., & Song, Z. (2013). Dynamic airspace sectorization via improved genetic algorithm. Journal of Modern Transportation J. Mod. Transport., 21(2), 117-124.

Delahaye, D., & Puechmorel, S. (2008). 3D airspace design by evolutionary computation. IEEE/AIAA 27th Digital Avionics Systems Conference.

Kulkarni, S., Ganesan, R., & Sherry, L. (2011). Static sectorization approach to dynamic airspace configuration using approximate dynamic programming. 2011 Integrated Communications, Navigation, and Surveillance Conference Proceedings.

Li, J., Wang, T., Hwang, I., & Hwang, I. (2009). A Spectral Clustering Based Algorithm for Dynamic Airspace Configuration. 9th AIAA Aviation Technology, Integration, and Operations Conference.

Ngueveu, S.U., Artigues, C., & Lopez P. (2016). Scheduling under a non-reversible energy source: An application of piecewise linear bounding of non-linear demand/cost functions. Discrete Applied Mathematics.

Sergeeva M., Delahaye D., Zerrouki L., & Schede N. (2015) Dynamic Airspace Configurationg Generated by Evolutionary Algorithms, DASC IEEE/AIAA 34thDigital Avionics Systems Conference.

Tang, J., Alam, S., Lokan, C., & Abbass, H. A. (2012). A multi-objective evolutionary method for Dynamic Airspace Re-sectorization using sectors clipping and similarities. 2012 IEEE Congress on Evolutionary Computation.

Trandac, H. & Duong, V.(2003). Optimised Sectorisation of Airspace with Constraints, Proceedings of the 5th Eurocontrol/FAA ATM R&D Seminar.

Xue, M. (2008). Airspace Sector Redesign Based on Voronoi Diagrams. AIAA Guidance, Navigation and Control Conference and Exhibit.